REMARKS

This Amendment is filed in response to the FINAL Office Action mailed July 21, 2009. A Request for Continued Examination and the associated fee is also filed herewith. All objections and rejections are respectfully traversed.

Claims 1-11, 17-34, and 36-41 are currently pending.

No new claims have been added.

Claims 1-3, 5-6, 9, 17, 19-26, 32, and 36-41 have been amended to better claim the invention.

Interview Summary

Applicant would like to thank Examiner Morrison for conducting the Applicant Initiated Interview on October 16, 2009 and for helping to advance this Application closer to allowance. Generally, as will be elaborated upon in greater detail below, the issue discussed involved Applicant's use of **removing from the hash table, *in response to determining that the hashed value of the selected entry of the first data set is in the hash table*, the hashed value of the selected entry of the first data set**. Specifically, Applicant discussed the differences between Applicant's hashed entries and Orwant's "members" and the overall difference between Applicant's claimed novel invention and the cited prior art. Examiner noted that a closer look at the prior art references would be conducted to verify Applicant's contentions and that another search would be completed. Examiner is encouraged to contact the undersigned attorney with any questions.

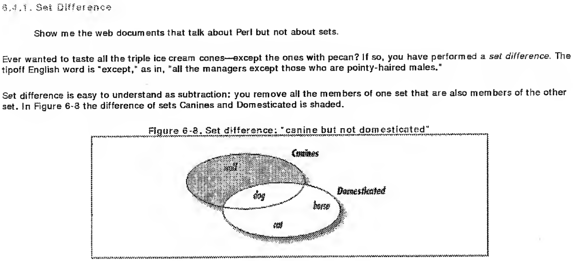
Rejections Under 35 U.S.C. §103

At paragraph 3 of the Office Action, claims 17-28 were rejected under 35 U.S.C. §103(a) as being unpatentable over Orwant et al., "*Mastering Algorithms with Perl*" (hereinafter "Orwant"), in view of Musser, "*Rationale for Adding Hash Table to the C++ Standard Template Library*" (hereinafter "Musser").

Applicant's claimed novel invention, as set forth in representative claim 22, comprises in part:

> 22. A method for comparing a first data set with a second data set, comprising:
>
> (a) selecting an entry from the first data set, wherein the first data set is stored on a source storage system;
>
> (b) determining if a hashed value of the selected entry of the first data set is in a hash table, wherein the hash table comprises one or more hashed values of the first data set;
>
> (c) adding, in response to determining that the hashed value of the selected entry of first data set is not in the hash table, the hashed value of the selected entry of the first data set to the hash table;
>
> (d) **removing from the hash table, *in response to determining that the hashed value of the selected entry of the first data set is in the hash table*, the hashed value of the selected entry of the first data set**;
>
> (e) selecting an entry from the second data set, wherein the second data set is stored on a destination storage system;
>
> (f) determining if a hashed value of the selected entry of the second data set is in the hash table, wherein the hash table further comprises one or more hashed entries of the second data set;
>
> (g) adding, in response to determining that the hashed value of the selected entry of the second data set is not in the hash table, the hashed value of the selected entry of the second data set to the hash table;
>
> (h) removing from the hash table, in response to determining that the hashed value of the selected entry of the second data set is in the hash table, the hashed value of the selected entry of the second data set;
>
> (i) continuing (a) through (d) and (e) through (h) respectively for all entries in the first and the second data sets until both the first and the second data sets have been completely processed; and
>
> (j) reporting a difference between the first data set and the second data set in response to at least one hashed value remaining in the hash table.

Orwant teaches, in relevant part as cited by Examiner, "set difference". In particular, Orwant explicitly describes set difference at section 6.4.1 as follows:



6.4.1. Set Difference

Show me the web documents that talk about Perl but not about sets.

Ever wanted to taste all the triple ice cream cones—except the ones with pecan? If so, you have performed a *set difference*. The tipoff English word is "except," as in, "all the managers except those who are pointy-haired males."

Set difference is easy to understand as subtraction: you remove all the members of one set that are also members of the other set. In Figure 6-8 the difference of sets Canines and Domesticated is shaded.

Figure 6-8. Set difference: "canine but not domesticated"

Orwant goes on further to explicitly define set difference as "noncommutative or asymmetric: that is, if you exchange the order of the sets, the result will change."

Musser teaches, in relevant part as cited by Examiner, a hash table (page 8, paragraph 9).

Applicant respectfully urges that Orwant, taken singly or in any combination with Musser, does not disclose Applicant's claimed novel and non-obvious use of

removing from the hash table, ***in response to determining that the hashed value of the selected entry of the first data set is in the hash table*****, the hashed value of the selected entry of the first data set.**

Applicant claims, in part, selecting an entry from a first data set and then determining if a hashed value of that selected entry is in a hash table, wherein the hash table comprises one or more hashed values from the first data set. In other words, broadly speaking, Applicant claims hashing entries of a first data set, storing one or more of the hashed values of the hashed entries in a hash table, and then determining if a particular hashed value of the first data set matches one of those stored hash values located in the hash table. ***In response to determining that the hashed value of the selected entry of the***

first data set is in the hash table, Applicant further claims **removing the hashed value of the selected entry of the first data set from the hash table.**

Applicant respectfully argues that Orwant does not teach Applicant's claimed novel **removing from the hash table,** ***in response to determining that the hashed value of the selected entry of the first data set is in the hash table*****, the hashed value of the selected entry of the first data set.** Specifically, Applicant respectfully notes that Orwant's "set difference" cannot be the same as Applicant's claimed first data set and/or second data set. In particular, Orwant explicitly states that "set difference" is noncommutative, meaning that if the order of the sets is changed, the results will also change. In contrast, Applicant's claimed data sets are commutative since, as stated in the claim language, hashed entries of the first data set (or the second data set) will be removed from the hash table ***in response to determining that a hashed value of the selected entry of the first data set*** (and/or a second data set) ***is in the hash table.*** In other words, because the response of removing an entry from a hash table is dependent upon determining if that hashed entry is in the hash table, and not dependent upon the order of the sets, the result will not change even if the two sets were switched. To further illustrate this point, step (i) in the claim language states continuing steps (a) through (d) and (e) through (h) respectively for all entries in the first and the second data sets until both the first and the second data sets have been completely processed. This shows that the outcome will not differ if the second data set is completed before the first data set. Thus, because Orwant's "set difference" is noncommutative, and because Applicant's claimed first and second data sets of representative claim 22 are commutative, Orwant must be silent to Applicant's claimed novel **removing from the hash table,** ***in response to determining that the hashed value of the selected entry of the first data set is in the hash table*****, the hashed value of the selected entry of the first data set.**

However, even if it is assumed *arguendo* that Orwant's "set difference" were the same as Applicant's claimed data set, Applicant respectfully argues that Orwant is still silent to Applicant's claimed novel **removing from the hash table,** ***in response to determining that the hashed value of the selected entry of the first data set is in the hash table*****, the hashed value of the selected entry of the first data set**. Specifically, Orwant explicitly states that "you remove all the members of one set that are also members of the other set." However, Orwant's "members" are not the same as Applicant's claimed entries because Orwant's "members" are not shown to be hashed. For example, Orwant describes a first member set as "canines" and a second member set as "not domesticated". Applicant respectfully points out that "canines" and/or "not domesticated" are not items shown to be hashed as they are merely group names. In contrast, Applicant claims removing a **hashed value of the selected entry of a first data set from a hash table** ***in response to determining that a hashed value of the selected entry of the first data set is in the hash table***. Thus, because Orwant's "members" are not shown to be hashed, Orwant must be silent to Applicant's claimed novel **removing from the hash table,** ***in response to determining that the hashed value of the selected entry of the first data set is in the hash table*****, the hashed value of the selected entry of the first data set**.

As noted above, Musser teaches, in relevant part as cited by Examiner, a hash table, however, Musser is silent to Applicant's claimed novel **removing from the hash table,** ***in response to determining that the hashed value of the selected entry of the first data set is in the hash table*****, the hashed value of the selected entry of the first data set**.

Accordingly, Applicant respectfully urges that Orwant, taken singly or in any combination with Musser, is legally insufficient to render the presently claimed invention obvious under 35 U.S.C. §103. Orwant and Musser, taken singly or in any combination, does not disclose Applicant's claimed novel and non-obvious use of **removing from the hash table,** ***in response to determining that the hashed value of the selected entry of the first data set is in the hash table*****, the hashed value of the selected entry of the first data set**.

One or more claims rejected by Examiner at paragraph 3 are dependent claims that are dependent from independent claims which are believed to be allowable for the reasons described above. Accordingly, those dependent claim(s) rejected by Examiner at paragraph 3 are believed to be in condition for allowance.

At paragraph 4 of the Office Action, claims 1-16, 29-31, and 39 were rejected under 35 U.S.C. §103(a) as being unpatentable over Orwant, in view of Musser, and in further view of Rsync, "*Rsync Unix command manual*" version 2.4.1 (hereinafter "Rsync").

Applicant's claimed novel invention, as set forth in representative claim 1, comprises in part:

> 1. A method for comparing a first directory comprising unique elements with a second directory comprising unique elements, comprising:
>
> (a) for each entry in the first directory, placing a hash value of the entry in a hash table, wherein the first directory is stored on a source storage system;
>
> (b) selecting an entry of the second directory, wherein the second directory is located on a destination storage system;
>
> (c) looking up a match between a hash value of the selected entry and the hash value of the entry in the hash table;
>
> (d) **removing, *in response to the match between the hash value of the selected entry and the hash value of the entry in the hash table*, the hash value of the entry from the hash table**;
>
> (e) determining if an additional second directory entry exists;
>
> (f) looping to step (b) in response to identifying the additional second directory entry; and
>
> (g) reporting a difference between the first directory and the second directory in response to at least one hash value entry remaining in the hash table.

Rsync teaches, in relevant part as cited by Examiner, a directory (OPTIONS section). However, Rsync is silent to Applicant's claimed novel **removing, *in response to the match between the hash value of the selected entry and the hash value of the entry in the hash table*, the hash value of the entry from the hash table**.

Additionally, as noted above, neither Orwant nor Musser disclose, teach, or suggest Applicant's claimed novel **removing from the hash table, *in response to determining that the hashed entry from the first data set is in the hash table*, the hashed entry from the first data set**. As such, because all independent claims comprises similar limitations not show by either prior art reference, Applicant respectfully urges that Rsync, taken singly or in any combination with Orwant and/or Musser, is legally insufficient to render the presently claimed invention obvious under 35 U.S.C. §103(a). Specifically, Rsync and/or Orwant and/or Musser, taken singly or in any combination, does not disclose, teach, or suggest Applicant's claimed novel and non-obvious use of **removing, *in response to the match between the hash value of the selected entry and the hash value of the entry in the hash table*, the hash value of the entry from the hash table**.

One or more claims rejected by Examiner at paragraph 4 are dependent claims that are dependent from independent claims which are believed to be allowable for the reasons described above. Accordingly, those dependent claim(s) rejected by Examiner at paragraph 4 are believed to be in condition for allowance.

At paragraph 5 of the Office Action, claims 32-34, 36-38, and 40-41 were rejected under 35 U.S.C. §103(a) as being unpatentable over Rsync, in view of Musser, and in further view of Orwant.

Applicant's claimed novel invention, as set forth in representative claim 37, comprises in part:

> 37. A computer readable medium containing executable program instructions executed by a processor, comprising:
>
> (a) program instructions that select an entry from a first data set, wherein the first data set is stored on a source storage system;
>
> (b) program instructions that determine if a hashed value of the selected entry of the first data set is in a hash table, wherein the hash table comprises one or more hashed values of the first data set;

(c) program instructions that add, in response to determining that the hashed value of the selected entry of first data set is not in the hash table, the hashed value of the selected entry of the first data set to the hash table;

(d) **program instructions that remove from the hash table, *in response to determining that the hashed value of the selected entry of the first data set is in the hash table*, the hashed value of the selected entry of the first data set**;

(e) program instructions that select an entry from a second data set, wherein the second data set is stored on a destination storage system;

(f) program instructions that determine if a hashed value of the selected entry of the second data set is in the hash table, wherein the hash table further comprises one or more hashed entries of the second data set;

(g) program instructions that add, in response to determining that the hashed value of the selected entry of the second data set is not in the hash table, the hashed value of the selected entry of the second data set to the hash table;

(h) program instructions that remove from the hash table, in response to determining that the hashed value of the selected entry of the second data set is in the hash table, the hashed value of the selected entry of the second data set;

(i) program instructions that continue (a) through (d) and (e) through (h) respectively for all entries in the first and the second data sets until both the first and the second data sets have been completely processed; and

(j) program instructions that report a difference between the first data set and the second data set in response to at least one hashed value remaining in the hash table

For similar reasons noted above, Applicant respectfully argues that neither Orwant nor Musser nor Rsync disclose, teach, or suggest Applicant's claimed novel **removing from the hash table, *in response to determining that the hashed value of the selected entry of the first data set is in the hash table*, the hashed value of the selected entry of the first data set**. As such, because all independent claims comprises similar limitations not show by either prior art reference, Applicant respectfully urges that Rsync, taken singly or in any combination with Orwant and/or Musser, is legally insufficient to render the presently claimed invention obvious under 35 U.S.C. §103(a). Specifically, Rsync and/or Orwant and/or Musser, taken singly or in any combination, does not disclose, teach, or suggest Applicant's claimed novel and non-obvious use of **program instructions that remove from the hash table,** ***in response to determining that the hashed value of the selected entry of the first data set is in the hash table*****, the hashed value of the selected entry of the first data set**.

One or more claims rejected by Examiner at paragraph 5 are dependent claims that are dependent from independent claims which are believed to be allowable for the reasons described above. Accordingly, those dependent claim(s) rejected by Examiner at paragraph 5 are believed to be in condition for allowance.

Applicant's Interpretation of the Prior Art

Applicant's interpretation of the prior art was derived, in part, from the following excerpts:

Orwant

> Set difference is noncommutative or asymmetric: that is, if you exchange the order of the sets, the result will change. (Section 6.4.1)

Musser

> There is another case in which hash tables are less set-like than sorted associative tables. (page 8, paragraph 9)

Rsync

> This tells rsync to copy directories recursively. (OPTIONS section)

Conclusion

All new claims and/or claim amendments are believed to be fully supported by Applicant's specification.

All independent claims are believed to be in condition for allowance.

All dependent claims are believed to be dependent from allowable independent claims, and therefore in condition for allowance.

Favorable action is respectfully solicited.

Please charge any additional fee occasioned by this paper to our Deposit Account No. 03-1237.

Respectfully submitted,

/Michael T. Abramson/

Michael T. Abramson
Reg. No. 60,320
CESARI AND MCKENNA, LLP
88 BLACK FALCON AVENUE
BOSTON, MA 02210
Telephone: (617) 951-2500
Facsimile: (617) 951-3927